Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Reporte final del proyecto de Integración:

Sistema Web para la identificación automática de aspectos académicos y de experiencia profesional en expedientes curriculares

"Proyecto Tecnológico"

Alumno
Ivan Alejandro Rosas Torres
208300244

Datos de asesor:
Maricela Claudia Bravo Contreras
Profesor Asociado
Departamento de Sistemas

Datos de Co-asesor.
José Alejandro Reyes Ortiz
Profesor Titular
Departamento de Sistemas

Trimestre 2015 Invierno

Fecha de entrega
10 de Abril de 2015

Yo, Maricela Claudia Bravo Contreras, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma del asesor

Yo, José Alejandro Reyes Ortiz, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma del asesor

Yo, Ivan Alejandro Rosas Torres, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Firma del alumno

RESUMEN

El proyecto de integración desarrollado es un sistema Web para la extracción de información académica y experiencia profesional de expedientes curriculares descritos en español utilizando reglas sintácticas y semánticas, consta de tres módulos que son: módulo de pre-procesamiento de texto, módulo de extracción de información y módulo de validación.

Módulo de pre-procesamiento de texto. En esta primera etapa se desarrolla la limpieza, la segmentación y el etiquetado de los expedientes curriculares. La limpieza quita todas las palabras que no aportan significado dentro de los expedientes curriculares. La segmentación envía una cadena de texto al módulo, está la dividirá en tokens o simplemente guarda cada palabra en una estructura de datos. En el etiquetado se determina la clase o categoría gramatical de cada palabra de una oración: palabra, número, puntuación, etc.

Módulo de extracción de información. Este módulo toma como entrada los expedientes curriculares pre-procesados con la finalidad de extraer la información de aspectos académicos y experiencia profesional mediante reglas semánticas y sintácticas, y dicha información extraída se almacena en una base de datos.

Módulo de validación. En este módulo se implementó un sistema web basado en el modelo vista controlador y un servidor de aplicaciones web. El sistema web es el encargado de ejecutar los módulos anteriores y desplegar en la interfaz la información extraída de aspectos académicos y experiencia profesional de los expedientes curriculares.

INDICE

INDICE DE FIGURAS

1. INTRODUCCIÓN

La Extracción de Información (EI) es una tarea que se encarga de localizar y extraer automáticamente información que complazca las necesidades de información de un usuario. La extracción de información se puede aplicar en los expedientes curriculares también llamado curriculum vitae.

El curriculum vitae es la relación ordenada de datos académicos, de formación y profesionales de una persona. Hay diversas empresas que almacenan los expedientes curriculares, en muchos casos se opta por almacenar dichos documentos de forma electrónica, este método ayuda mucho cuando se quiere reducir los grandes volúmenes de papel que dicho proceso puede generar.

En la actualidad para una persona es muy difícil encontrar de manera manual información útil de los curriculums vitae, ya que el volumen de estos textos puede ser muy grande; dicha persona utilizará mucho tiempo y será tedioso el obtener datos relevantes ya sea de la formación académica o experiencia profesional.

Por lo tanto, en este proyecto se propone extraer la información académica y experiencia profesional de los curriculums vitae descritos en español utilizando técnicas de Procesamiento de Lenguaje Natural (PLN), específicamente reglas sintácticas y semánticas. La información extraída será validada por un usuario en una interfaz web para su inserción en una base de datos.

2. ANTECEDENTES

2.1. Referencias Internas

Extracción automatizada y representación de servicios Web mediante ontologías [1]

En este proyecto se extrae la información más importante de cada una de las secciones de los archivos de descripción de servicios Web y que sean

representados mediante ontologías. El proyecto propuesto también extraerá información de un conjunto de archivos, utilizando reglas sintácticas y semánticas.

La diferencia que existe es que, en el proyecto propuesto la extracción de información es sobre aspectos académicos y experiencia profesional de un conjunto de expedientes curriculares en formato PDF.

Sistema de recuperación de información semántica [2]

El proyecto aplica técnicas de procesamiento de lenguaje natural para la recuperación de información semántica. La idea general es similar que el proyecto de integración propuesto: tomar varios archivos de texto, procesarlos y extraer información relevante para su validación e inserción en una base de datos.

La diferencia que existe es que, en este proyecto se extrae información específica de artículos de investigación, por su parte, en el proyecto propuesto se hace la extracción de información de aspectos académicos y experiencia profesional a partir de curriculums vitae logrando una comprensión semántica de los textos.

Sistema de almacenamiento semántico y recuperación de textos de investigación mediante ontologías [3]

En este proyecto trabaja con Sistema de Recuperación de Información Semántica (SRIS) para la recuperación de información de textos de investigación y almacenar dicha información mediante base de datos y ontologías. Mientras que, el proyecto propuesto extraerá información relevante de los expedientes curriculares para su validación e inserción en una base de datos.

La diferencia que existe es que, en este proyecto de integración se utilizarán reglas sintácticas y semánticas para la extracción de información de los expedientes curriculares.

2.2. Referencias Externas

Extracción automática de información semántica basada en estructuras sintácticas [4]

En esta tesis se utilizó la extracción de información semántica a partir de libros didácticos, se eligen temas de interés y de estas se crea un conjunto de oraciones. Se realiza un análisis sintáctico y se extraen las palabras de cada oración para después almacenarlos en una base de datos. El proyecto propuesto utilizará

igualmente la extracción de información pero aplicado a expedientes curriculares para obtener la formación académicos y experiencia profesional para después en una interfaz se enfatizarán dichos datos de los expedientes curriculares.

Learning to Automatically Solve Algebra Word Problems [5]

En este trabajo de investigación se obtuvo un sistema para la extracción de información relevante para resolver automáticamente problemas de algebra con una estructura de lenguaje natural. El proyecto propuesto utilizaráotras reglas para realizar el análisis semántico y la extracción de información de archivos de texto en lenguaje natural y el objetivo a alcanzar es distinto. En el artículo de investigación se quiere resolver problemas de algebra, en el proyecto propuesto el objetivo es ayudar a la identificación de información sobre la formación académica y experiencia profesional a partir de un gran número de expedientes curriculares.

AntConc [6]

Es un software gratuito para el análisis estadístico de textos. Este software puede leer un archivo de texto (txt, htm, html, xml) en inglés y lleva acabo la búsqueda del término deseado y mostrara, todos los resultados posibles: concordancias y expresiones que se relacionen de alguna manera con el término de interés. El proyecto propuesto obtendrá y validará, la información académica y experiencia profesional de archivos de texto en formato PDF en español.

3. JUSTIFICACIÓN

Actualmente la tecnología moderna otorga grandes volúmenes de información en donde muchas veces los textos están escritos en lenguaje natural, con la extracción de información en este caso de expedientes curriculares se puede obtener datos que son de interés para una aplicación específica.

Esta propuesta de proyecto de integración se basará en la extracción de información de aspectos relacionados con la formación académica y experiencia profesional. Además, se realizará la implementación de un sistema web en donde se pueda realizar una extracción, validación e inserción de la información antes mencionada.

Este sistema beneficiará en tener un orden de la información de los expedientes curriculares y de esa forma ayudar a que el usuario pueda obtener datos relevantes de dichos expedientes en un tiempo considerable. Los beneficiarios pueden ser empresa u organizaciones que tienen un gran volumen de curriculums y de esa forma facilitar las consultas de información que sea de su interés.

Al ser una implementación con reglas sintácticas y semánticas de Procesamiento de Lenguaje Natural en un sistema web es un proyecto que puede facilitar la organización de expedientes curriculares para la contratación de personal en una institución educativa, empresa privada o institución gubernamental.

4. OBJETIVOS

4.1. Objetivo General

Diseñar e implementar un sistema Web para la extracción automática de información relevante sobre aspectos académicos y de experiencia profesional a partir de expedientes curriculares, descritos en español utilizando reglas sintácticas y semánticas.

4.2. Objetivos Específicos:

- Diseñar e implementar un módulo de software para el procesamiento de textos en español de expedientes curriculares con una estructura de lenguaje natural.
- Desarrollar una base de conocimiento con reglas sintácticas y semánticas para identificar la información académica y experiencia profesional a partir de expedientes curriculares.
- Diseñar e implementar una arquitectura web para la ingeniería de textos que utilice la base de conocimiento con la finalidad de realizar el proceso completo de extracción de la información.
- Diseñar e implementar un módulo para la validación, por parte del usuario, de la información extraída y su almacenamiento en una base de datos.

5. MARCO TEÓRICO

5.1. Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural es una sub-rama de la Inteligencia Artificial y de la Lingüística. También se suele referir a esta rama de la informática de forma abreviada como PLN o NLP, del inglés Natural LanguageProcessing.

Es una disciplina, originalmente desarrollado a comienzos de la Guerra Fría como el mecanismo que usaban los físicos Soviéticos para la traducción de documentos, es uno de los primeros objetivos computacionales más investigados. Estos esfuerzos prematuros, por analizar y modelar el lenguaje humano, fueron caracterizados por una técnica sin conocimiento lingüístico y por el bajo rendimiento computacional de la época.

El fin del PLN es construir sistemas y mecanismos que permitan la comunicación entre personas y máquinas por medio de lenguajes naturales ya sea por medio de la voz o del texto. Además, trata de que los mecanismos que permitan esa comunicación sean lo más eficaces posibles, computacionalmente hablando. En definitiva, se busca poder crear programas que puedan analizar, entender y generar lenguajes que los humanos utilizan habitualmente, de manera que el usuario pueda llegar a comunicarse con la computadora de la misma forma que lo haría con un humano.

5.1.1. Niveles en el Procesamiento del Lenguaje Natural

Las técnicas del procesamiento del lenguaje se promueven a través de diferentes análisis ocupando cada uno de ellos distintos niveles:

Análisis morfológico. Efectúa un examen de cada vocablo para obtener toda la información gramatical de la misma como prefijos, raíces y sufijos, así como la clase gramatical o clases a las que pertenece.

Análisis sintáctico. Tiene por objeto comprobar si los vocablos del texto están bien coordinados y unidos, es decir, averiguar si las oraciones son gramaticalmente correctas. Además, en este estudio se pretende también la resolución de problemas no solucionados en el análisis anterior como la reiterada

ambigüedadgramatical de las palabras, destinando para ello varios mecanismos como las posiciones de las voces en las oraciones o los contrastes gramaticales.

Análisis semántico. La semántica estudia la significación de las palabras, por tanto, un análisis semántico tratará de averiguar el significado de las oraciones de un texto, y por extensión el del mismo texto.

5.1.2. Problemas en el uso del Lenguaje Natural

Los sistemas de PLN deben atacar una variedad de problemas relacionados con el lenguaje natural:

- Inexactitud: incluyendo errores ortográficos, signos de puntuación incorrectos, palabras transpuestas, y oraciones agramaticales.
- Imprecisión: incluyendo el uso de términos relativos sin un punto específico de referencia y el uso de términos cualitativos.
- Ambigüedad: debido a que pueden surgir múltiples interpretaciones en cualquier nivel del conocimiento lingüístico.

5.2. World Wide Web

En poco más de una década desde su aparición, la World Wide Web se ha convertido en un instrumento de uso cotidiano en nuestra sociedad, comparable a otros medios tan importantes como la radio, la televisión o el teléfono, a los que aventaja en muchos aspectos. La web es hoy un medio extraordinariamente flexible y económico para la comunicación, el comercio y los negocios, ocio y entretenimiento, acceso a información y servicios, difusión de cultura, etc. Paralelamente al crecimiento espectacular de la web, las tecnologías que la hacen posible han experimentado una rápida evolución. Desde las primeras tecnologías básicas: HTML y HTTP, hasta nuestros días, han emergido tecnologías como CGI, Java, JavaScript, ASP, JSP, PHP, Flash, J2EE, XML, por citar algunas de las más conocidas, que permiten una web mejor, más amplia, más potente, más flexible, o más fácil de mantener. Estos cambios influyen y son al tiempo influidos por la propia transformación de lo que entendemos por WWW. La generación dinámica de páginas, el acoplamiento con bases de datos, la mayor interactividad con el usuario, la concepción de la web como plataforma universal para el despliegue de

aplicaciones, la adaptación al usuario, son algunas de las tendencias evolutivas más marcadas de los últimos años.

5.2.1. Un poco de historia

La aparición de la WWW se puede situar en 1989, cuando Tim Berners-Lee presentó su proyecto de “World Wide Web” en el Consejo Europeo para la Investigación Nuclear, con las características esenciales que perduran en nuestros días. El propio Berners-Lee completó en 1990 el primer servidor web y el primer cliente, y un año más tarde publicó el primer borrador de las especificaciones de HTML y HTTP. El lanzamiento en 1993 de Mosaic, el primer navegador de dominio público, compatible con Unix, Windows, y Macintosh, por el National Center forSupercomputingApplications (NCSA), marca el momento en que la WWW se da a conocer al mundo, extendiéndose primero en universidades y laboratorios, y en cuestión de meses al público en general, iniciando el que sería su vertiginoso crecimiento. Los primeros usuarios acogieron con entusiasmo la facilidad con que se podían integrar texto y gráficos y saltar de un punto a otro del mundo en una en una misma interfaz, y la extrema sencillez para contribuir contenidos a una web mundial.

5.3. Extracción de Información

La extracción de información analiza un conjunto de de textos, luego los transforma en información que es procesada y analizada. Esto identifica los fragmentos de textos relevantes, extrae la información relevante de los fragmentos, y con estas piezas organiza la información requerida en una estructura coherente. Se trata de reconocer la información importante contenida en los documentos y trasladarla a un formato predefinido para que pueda ser tratada y recuperada con mayor facilidad. Además de la información relevante deben conseguirse las relaciones entre ellos, mientras que se ignora la información irrelevante. Hoy día, sin embargo, los sistemas de extracción de información tratan solamente con tipos específicos de textos y solo tienen buenos resultados en algunos componentes.

6. DESARROLLO DEL PROYECTO

A continuación se explica el desarrollo del sistema web, los módulos que fueron implementados están esquematizados en la Figura 1.

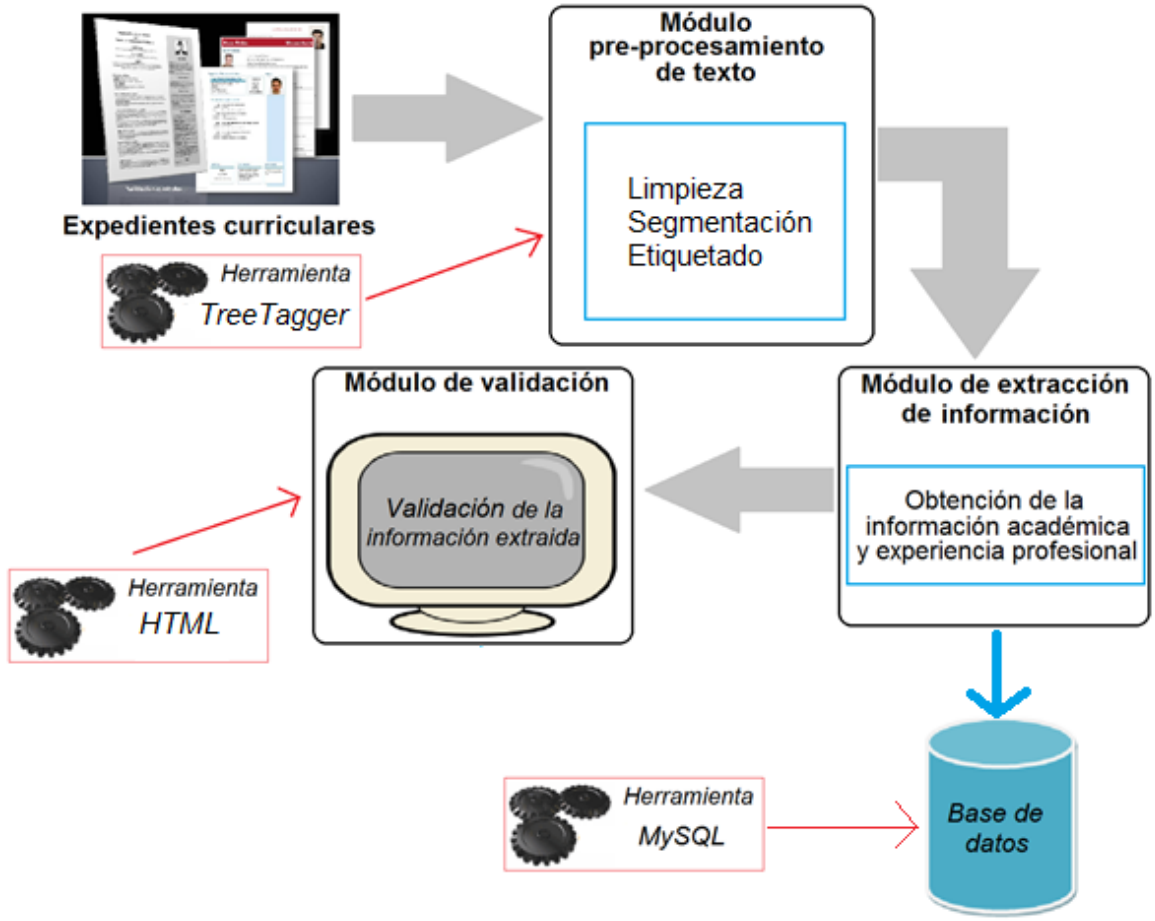


Figura 1. Arquitectura del Sistema Web de extracción de información académica y experiencia profesional en expedientes curriculares

Antes de implementar el módulo de pre-procesamiento de texto los expedientes curriculares en formato pdf, se tienen que convertir en formato txt para facilitar la manipulación de la información contenida, el código se puede ver en el anexo 9.1.

6.1. Elaboración del módulo de pre-procesamiento de texto

Esta es la primera etapa que se desarrolló del sistema, se compone de tres fases, la limpieza de los expedientes curriculares, la segmentación y finalmente el etiquetado.

6.1.1. Limpieza de los expedientes curriculares

Los expedientes curriculares obtenidas de un corpus se envían a procesar en la fase de limpieza, esta fase se encarga de buscar dentro de un listado de palabras vacías[1] todas las coincidencias existentes dentro de los expedientes curriculares, al encontrar alguna de estas palabras que no aporta significado relevante las sustituye en los expedientes curriculares por un espacio en blanco. El conjunto de palabras vacías utilizadas en esta fase se muestran en la Figura 2. Una vez que el módulo de limpieza quita todas las palabras sin significado que encuentre dentro del texto, da como salida un texto limpio de los expedientes curriculares, en la Figura 3 podemos ver un ejemplo de lo que realiza esta fase del módulo de pre-procesamiento de texto, el código se puede ver en el anexo 9.2.

```
acuerdo,adelante,ademas,además,adrede,ahi,ahí,ahora,al,alli,allí,alrededo
r,antano,antaño,antes,apenas,aproximadamente,aquel,aquél,aquella,aquélla,
aquellas,aquéllas,aquello,aquellos,aquéllos,aqui,aquí,arriba,asi,así,aun,
aún,aunque,bajo,bastante,bien,breve,casi,cerca,claro,como,cómo,con,conmig
o,contigo,contra,cual,cuál,cuales,cuáles,cuando,cuándo,cuanta,cuánta,cuan
tas,cuántas,cuanto,cuánto,cuantos,cuántos,debajo,delante,demasiado,dentro
,deprisa,despacio,despues,después,detras,detrás,dia,día,dias,días,donde,d
ónde,dos,durante,él,ella,ellas,ellos,encima,enfrente,enseguida,entre,es,e
sa,ésa,esas,ésas,ese,ése,eso,esos,ésos,está,ésta,estado,estados,estan,est
án,estar,estas,éstas,éste,esto,estos,éstos,ex,excepto,final,fue,fuera,fue
ron,g,general,gran,ha,habia,había,habla,hablan,hace,hacia,han,hasta,hay,h
oras,hoy,incluso,informo,informó,junto,lado,las,le,lejos,lo,los,luego,mal
,mas,más,mayor,me,medio,mejor,menos,menudo,mi,mí,mia,mía,mias,mías,mientr
as,mio,mío,mios,míos,mis,mismo,mucho,muy,n,nada,nadie,ninguna,no,nos,noso
tras,nosotros,nuestra,nuestras,nuestro,nuestros,nueva,nuevo,nunca,os,otra
,otros,pais,país,parte,pasado,peor,pero,poco,p_or,porque,pronto,proximo,p
róximo,puede,que,qué,quien,quién,quienes,quiénes,quiza,quizá,quizas,quizá
s,raras,repente,salvo,se,sé,segun,según,ser,sera,será,si,sí,sido,siempre,
sin,so_bre,solamente,solo,sólo,son,soyos,su,supuesto,sus,suya,suyas,suyo,
t,tal,tambien,también,tampoco,tarde,te,temprano,ti,tiene,todavia,todavía,
todo,todos,tras,tu,tú,tus,tuya,tuyas,tuyo,tuyos,u,un,una,unas,uno,unos,us
ted,ustedes,v,veces,vez,vosotras,vosotros,vuestra,vuestras,vuestro,vuestr
os,w,x,y,ya,yo,z
```

Figura 2. Conjunto de palabras vacías

[1] Palabras vacías: nombre que reciben las palabras sin significado como artículos, pronombres, preposiciones, etc.

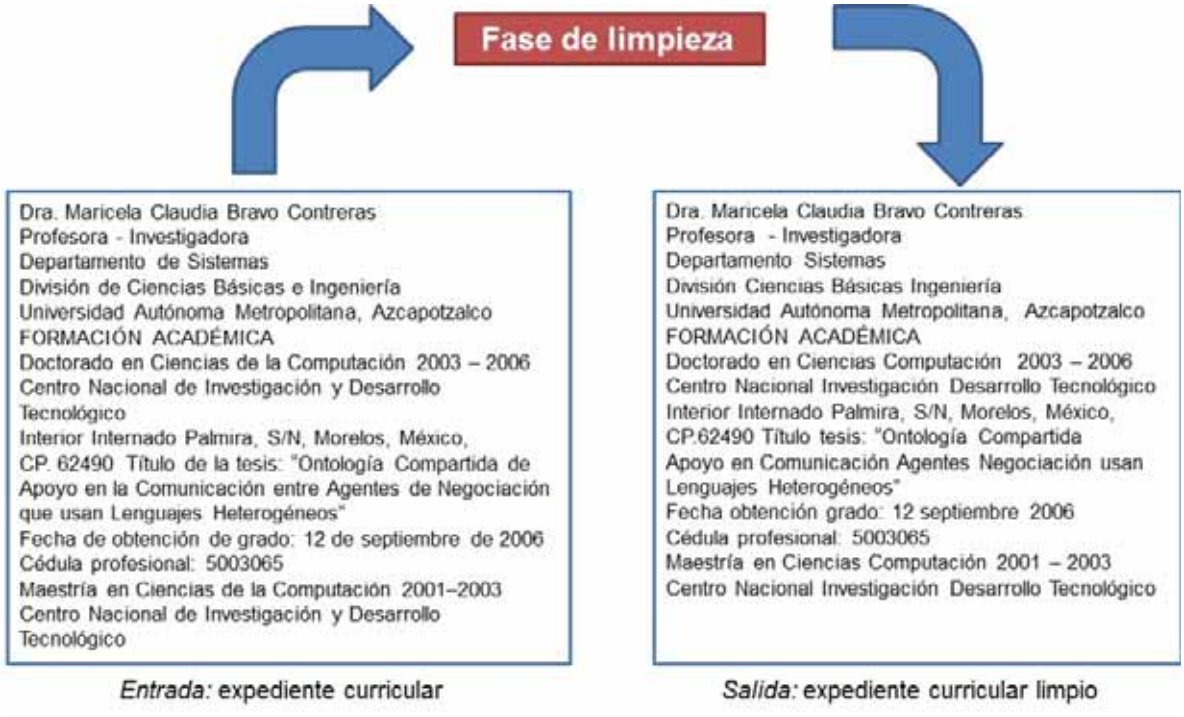


Figura 3. Fase de limpieza de expedientes curriculares

6.1.2. Segmentación de los expedientes curriculares

En esta fase se implementa con la ayuda de la herramienta TreeTagger[2], al enviarle una cadena de texto al módulo, está la divide en tokens o simplemente guarda cada palabra en una estructura de datos, esto es quitar los espacios dentro del expediente curricular y segmentarla por palabras, la salida de este módulo es un arreglo de cadenas donde se almacenan todas las palabras que contiene el expediente curricular limpio, en la Figura 4 podemos ver un ejemplo de lo que realiza la fase de segmentación, el código se puede ver en el anexo 9.3.

[2]TreeTagger es una herramienta para la anotación de texto con la parte de expresión y de información lema.

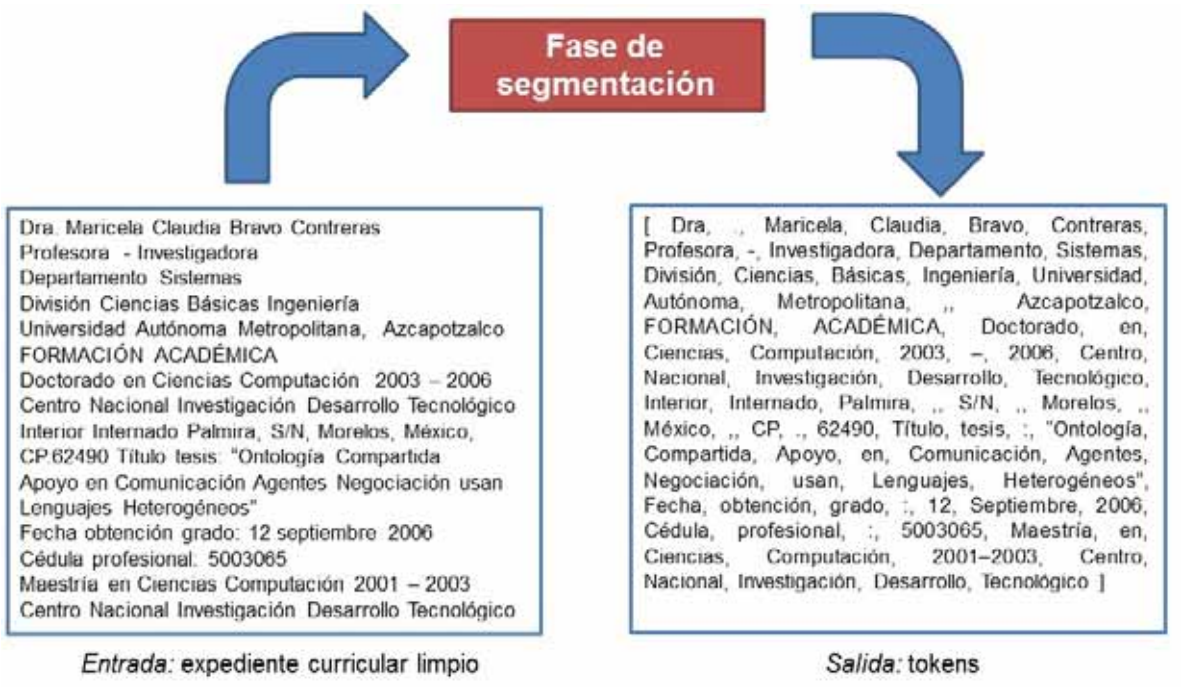


Figura 4. Fase de segmentación de expedientes curriculares

6.1.3. Etiquetado de los expedientes curriculares

La última fase del módulo de pre-procesado de texto es el etiquetado, este se desarrolló de la siguiente forma: una vez que se recibe de la fase anterior el expediente curricular segmentado en palabras, se utiliza la herramienta TreeTagger ya que gracias a sus funciones podemos realizar un análisis morfológico que consiste en determinar la clase o categoría gramatical de cada palabra de una oración: palabra, número, puntuación, etc. Una vez etiquetadas las palabras estas se guardan en un arreglo de cadenas, donde tendremos el pos (clasificación de las palabras). La salida de este módulo serán varias cadenas de palabras clave pre-procesadas y etiquetadas gramaticalmente, en la Figura 5 podemos ver un ejemplo de lo que realiza la fase de etiquetado, el código se puede ver en el anexo 9.3.

Figura 5. Fase de etiquetado de expedientes curriculares

6.2. Elaboración del módulo de extracción de información

Este módulo es el encargado de extraer la información de aspectos académicos y experiencia profesional, para llevar a cabo esta tarea se implementó en java mediante expresiones regulares basadas en técnicas de Procesamiento de Lenguaje Natural, específicamente reglas semánticas y sintácticas.

El módulo de extracción recibe de entrada las palabras procesadas de los expedientes curriculares obtenidos del módulo de pre-procesamiento de texto, se utilizan métodos de PLN en específicamente reglas semánticas y sintácticas para poder encontrar la información académica y experiencia profesional. La salida son las frases o palabras donde tengamos la información académica y la información de la experiencia profesional que son almacenadas en una base de datos.

6.2.1. Extracción de información sobre aspectos académicos

Para llevar a cabo la implementación de esta parte del módulo, es necesario implementar reglas semánticas y sintácticas, con el objetivo de encontrar una estructura que nos pudiera referenciar donde se encuentra la información académica, el código se puede ver en el anexo 9.4.

La extracción de la información académica se realiza con la aplicación de los siguientes patrones:

Patrones

P1.1: (token=Licenciatura | Licenciado)& (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP) & (pos=NC | NP)

De no encontrar P1.1 buscar P1.2

P1.2: (token= Licenciatura | Licenciado) & (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP)

De no encontrar P1.2 buscar P1.3

P1.3: (token= Licenciatura | Licenciado) & (token=PREP = en) & (pos=NC | NP)

P2.1: (token= Maestría | Maestro) & (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP)& (pos=NC | NP)

De no encontrar P2.1 buscar P2.2

P2.2: (token= Maestría | Maestro) & (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP)

De no encontrar P2.2 buscar P2.3

P2.3: (token= Maestría | Maestro) & (token=PREP = en) & (pos=NC | NP)

P3.1: (token= Doctorado | Doctor) & (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP) & (pos=NC | NP)

De no encontrar P3.1 buscar P3.2

P3.2: (token= Doctorado | Doctor) & (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP)

De no encontrar P3.2 buscar P3.3

P3.3: (token= Doctorado | Doctor) & (token=PREP = en) & (pos=NC | NP)

P4.1: (token= Ingeniería | Ingeniero) & (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP) & (pos=NC | NP)

De no encontrar P4.1 buscar P4.2

P4.2: (token= Ingeniería | Ingeniero) & (token=PREP = en) & (pos=NC | NP) & (pos=NC | NP)

De no encontrar P4.2 buscar P4.3

P4.3: (token= Ingeniería | Ingeniero) & (token=PREP = en) & (pos=NC | NP)

El resultado de la extracción después de aplicar los patrones de informaciónacadémicos, se muestran en la figura 6.

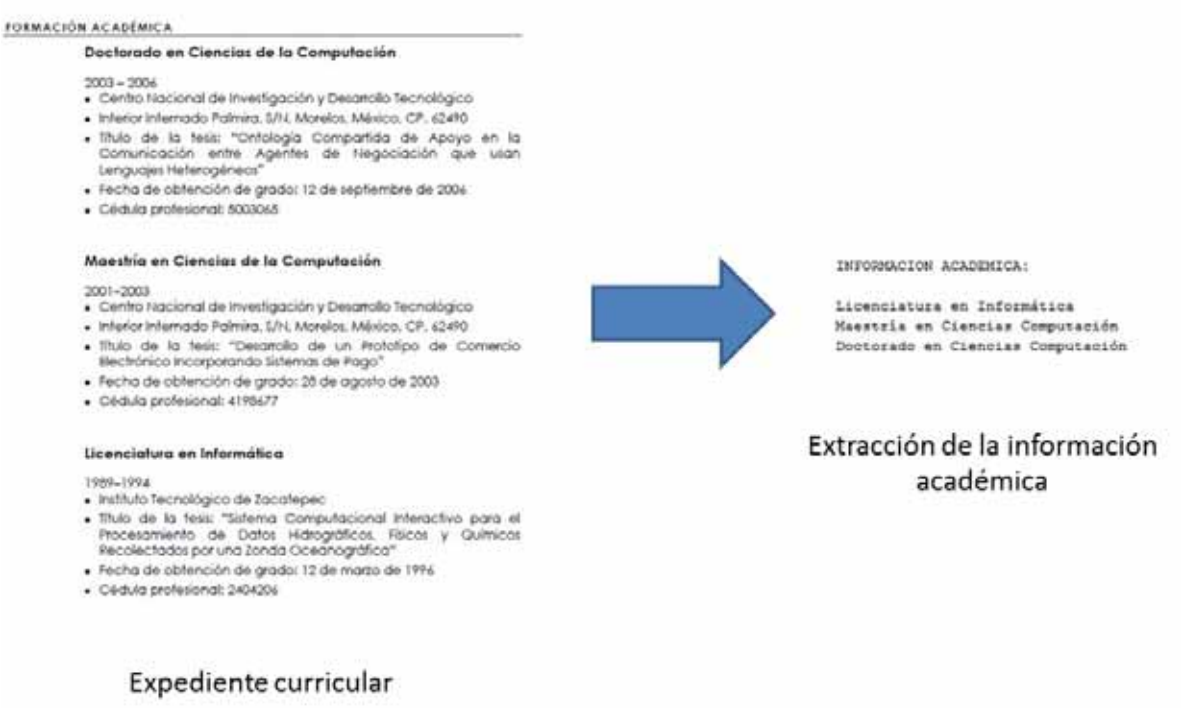


Figura 6. Extracción de la información académica

6.2.2. Extracción de información de la experiencia profesional

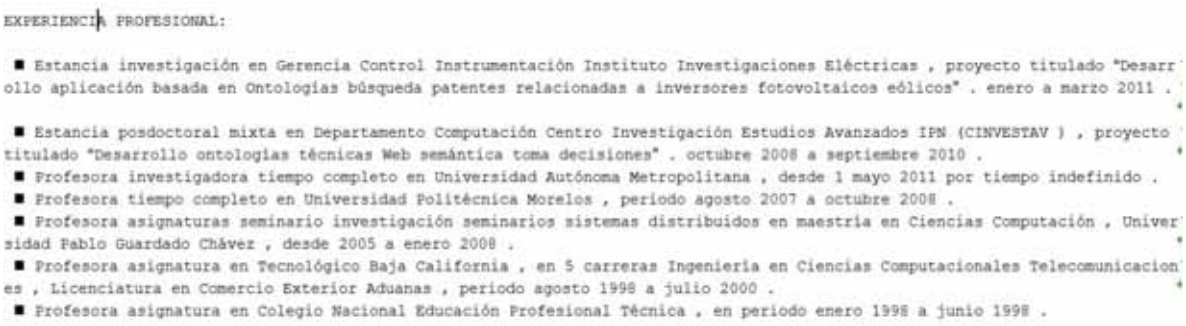
Para llevar a cabo la implementación de la segunda parte de este módulo, es necesario implementar reglas semánticas y sintácticas, con el objetivo de encontrar una estructura que nos pudiera referencias donde se encuentra la información de la experiencia profesional, el código se puede ver en el anexo 9.5.

La extracción de la experiencia profesional se realiza con la aplicación de los siguientes patrones:

Patrones

P1:(token= ".") & [(pos= CODE) | (pos=CARD)]
P2: (token= ".")&[(pos= ADJ) | (token= "indefinido")]
"indefinido" se refiere al tiempo de duración de la experiencia profesional
P3: (token= ".")&(pos= NP)&[(token= ",") | (token= ".")]
P4: (pos= "RP") &(pos= VLadj)

El resultado de la extracción después de aplicar los patrones de la experiencia profesional, se muestran en la figura 7.



EXPERIENCIA PROFESIONAL:

- Estancia investigación en Gerencia Control Instrumentación Instituto Investigaciones Eléctricas , proyecto titulado "Desarrollo aplicación basada en Ontologías búsqueda patentes relacionadas a inversores fotovoltaicos eólicos" . enero a marzo 2011 .
- Estancia posdoctoral mixta en Departamento Computación Centro Investigación Estudios Avanzados IPN (CINVESTAV) , proyecto titulado "Desarrollo ontologías técnicas Web semántica toma decisiones" . octubre 2008 a septiembre 2010 .
- Profesora investigadora tiempo completo en Universidad Autónoma Metropolitana , desde 1 mayo 2011 por tiempo indefinido .
- Profesora tiempo completo en Universidad Politécnica Morelos , periodo agosto 2007 a octubre 2008 .
- Profesora asignaturas seminario investigación seminarios sistemas distribuidos en maestría en Ciencias Computación , Universidad Pablo Guardado Chávez , desde 2005 a enero 2008 .
- Profesora asignatura en Tecnológico Baja California , en 5 carreras Ingeniería en Ciencias Computacionales Telecomunicaciones , Licenciatura en Comercio Exterior Aduanas , periodo agosto 1998 a julio 2000 .
- Profesora asignatura en Colegio Nacional Educación Profesional Técnica , en periodo enero 1998 a junio 1998 .

Figura 7. Extracción de la experiencia profesional

6.3. Base de Datos

Para almacenar la información académica y experiencia profesional se desarrolló una base de datos, en la Figura 8 se muestra el diagrama de la base de datos implementada, el código en donde se implementan las consultas se puede ver en el anexo 9.6.

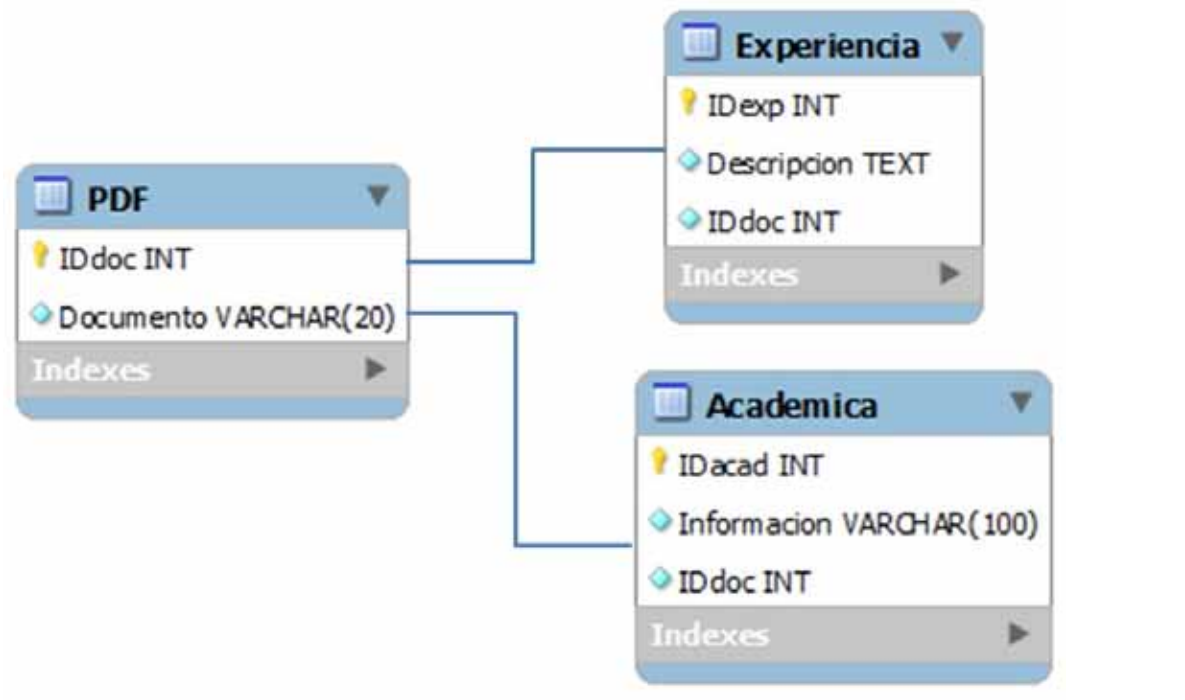
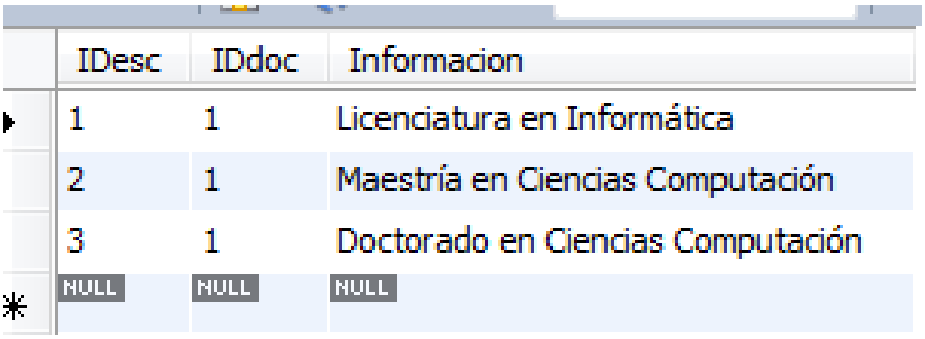


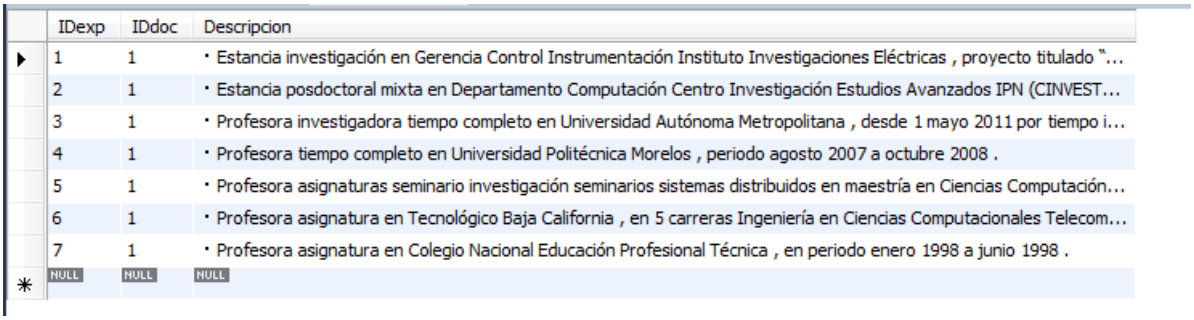
Figura 8. Diagrama de la base de datos

La base de datos se implementa con la ayuda de la herramientaMySQL, el almacenamiento de la información académica se muestra con una tabla de nombreAcadémica (Figura 9) y el almacenamiento de la experiencia profesional se muestra con una tabla de nombre Experiencia (Figura 10).



IDesc	IDdoc	Informacion
1	1	Licenciatura en Informática
2	1	Maestría en Ciencias Computación
3	1	Doctorado en Ciencias Computación
NULL	NULL	NULL

Figura 9. Tabla donde se almacena la información académica



IDexp	IDdoc	Descripcion
1	1	· Estancia investigación en Gerencia Control Instrumentación Instituto Investigaciones Eléctricas , proyecto titulado "...
2	1	· Estancia posdoctoral mixta en Departamento Computación Centro Investigación Estudios Avanzados IPN (CINVEST...
3	1	· Profesora investigadora tiempo completo en Universidad Autónoma Metropolitana , desde 1 mayo 2011 por tiempo i...
4	1	· Profesora tiempo completo en Universidad Politécnica Morelos , periodo agosto 2007 a octubre 2008 .
5	1	· Profesora asignaturas seminario investigación seminarios sistemas distribuidos en maestría en Ciencias Computación...
6	1	· Profesora asignatura en Tecnológico Baja California , en 5 carreras Ingeniería en Ciencias Computacionales Telecom...
7	1	· Profesora asignatura en Colegio Nacional Educación Profesional Técnica , en periodo enero 1998 a junio 1998 .
NULL	NULL	NULL

Figura 10. Tabla donde se almacena la experiencia profesional

6.4. Elaboración del módulo de validación de la información

Este módulo es el encargado de ejecutar los módulos anteriores y desplegará en un sistema web la información extraída de aspectos académicos y experiencia profesional de los expedientes curriculares, para llevar a cabo esta tarea se implementó en java con tecnologías web (HTML, Servlets) basadas en el modelo vista controlador y un servidor de aplicaciones web (Tomcat), el código se puede ver en el anexo 9.7.

A continuación se muestra las capturas del sistema web. La Figura 11 nos muestra la interfaz inicial del sistema.

Figura 11. Interfaz inicial del sistema web

En la Figura 12 nos muestra que al hacer click en el botón Iniciar se da el comienzo de la ejecución del sistema. Para cargar los expedientes curriculares hacemos click en el botón Buscar para buscar los archivos a procesar, esta acción nos abrirá el explorador de archivos de nuestro sistema operativo, de tal manera que podamos navegar hasta la ruta donde se encuentran los expedientes curriculares, posteriormente seleccionamos el o los expedientes curriculares que se subirán al sistema y damos click en el botón Abrir, así como lo muestra la Figura 13. En el caso de no ser extensión .pdf mandara el sistema un mensaje de archivo no valido como se muestra en la Figura 14. Una vez que presionamos el botón abrir se nos mostraran en el sistema los archivos seleccionados y procedemos a subir los archivos al servidor, esto se realiza al dar click en el botón Subir como se muestra en la Figura 15 y se cargaran los archivos al servidor.

Figura 12. Iniciar el proceso de extracción de información

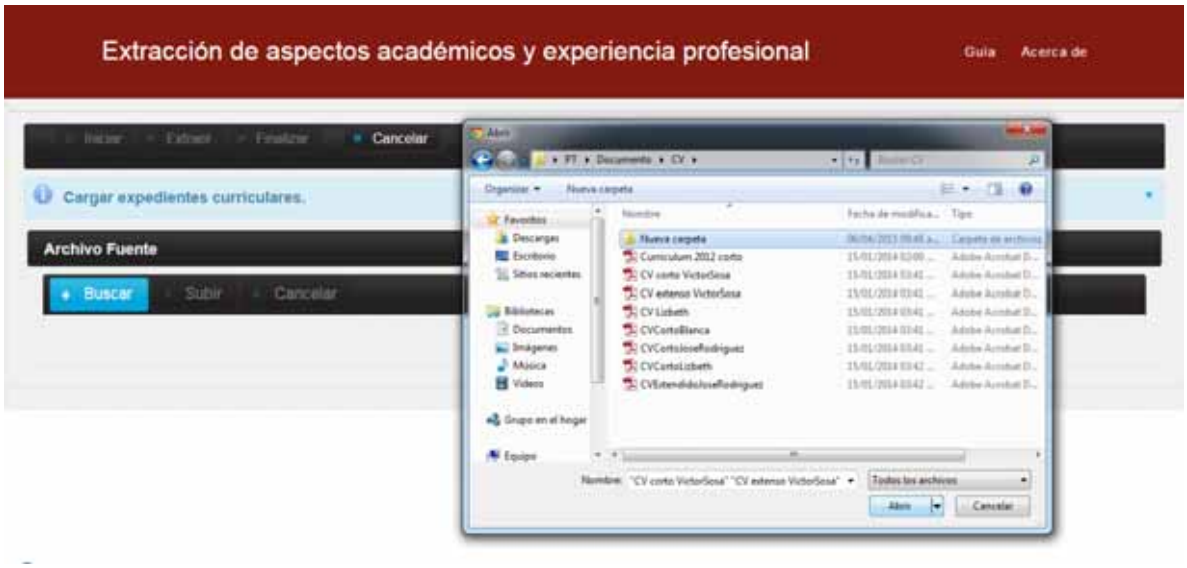


Figura 13. Selección de expedientes curriculares

Figura 14. Mensaje de archivo no valido

Figura 15. Subir expediente curricular al servidor

La Figura 16 se muestra una lista de los nombres de los expedientes curriculares que se subieron al servidor y el total de los mismos; si se requiere subir más expedientes curriculares volvemos a realizar los pasos anteriores, es decir, buscar, seleccionar y subir expedientes curriculares.

Figura 16. Expediente curricular cargado al servidor

Una vez realizado la carga de los expedientes curriculares procedemos a extraer la información de aspectos académicos y experiencia profesional, para realizar este proceso damos click en el botón Extraer. El proceso de Extraer implica la limpieza, segmentación, etiquetado, extracción de la información académica, extracción de la experiencia profesional y guarda la información en la base de datos; al terminar el proceso se visualiza en la interfaz la información de aspectos académicos y experiencia profesional de los expedientes curriculares como se muestra en la Figura 17.

Figura 17. Visualización de la información académica y experiencia profesional

7. RESULTADOS

El conjunto de archivos para realizar las pruebas fue un corpus de 12 expedientes curriculares en idioma español, estos expedientes curriculares tienes una extensión .pdf que es un formato de almacenamiento de documentos digitales. En la Figura 18 podemos ver el corpus de expedientes curriculares.

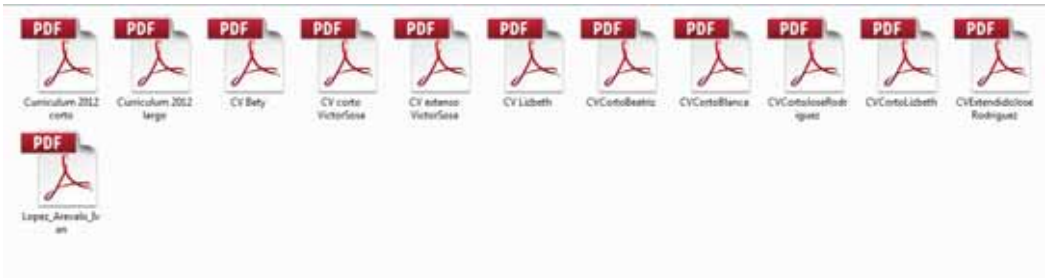


Figura 18. Corpus de expedientes curriculares

El tiempo de procesamiento estimado que tardó el sistema en analizar, extraer y guardar en la base de datos, para un corpus de 12 expedientes curriculares en idioma español fue de 41 segundos y se puede corroborar en la Figura 19.

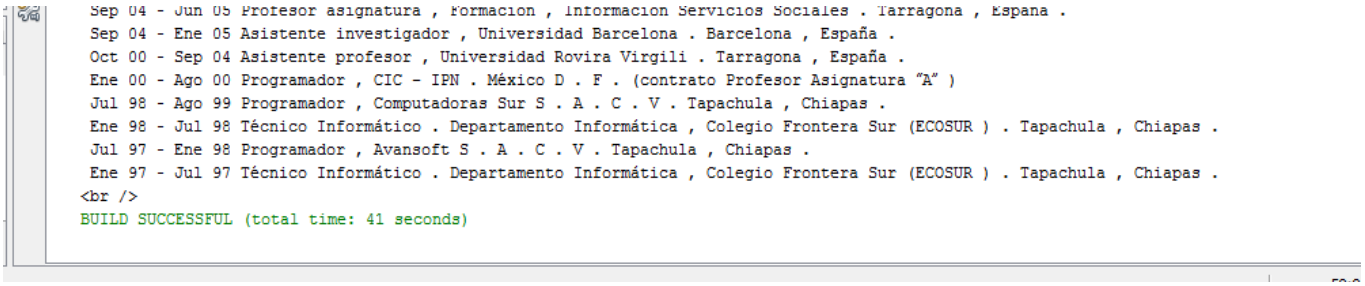


Figura 19. Tiempo estimado de procesamiento

7.1. Resultados de la extracción de aspectos académicos

Para la evaluación del proceso de extracción de información sobre aspectos académicos a partir de expedientes curriculares, tenemos que analizar la precisión con la que se obtuvieron los resultados, se tiene que revisar el número de aspectos académicos y el número de aspectos académicos erróneos. En la Figura 20 podemos encontrar los datos erróneos extraídos estos nos dan un número total de 3 y el número total de aspectos académicos es de 27. Con la ayuda de estos datos a continuación calcularemos la precisión del sistema web.

$$\boldsymbol{Elementos\ Erroneos\ Extraidos = 3}$$

$$\boldsymbol{Elementos\ Relevantes\ Extraidos = 24}$$

$$Precisión\ = \frac{elementos\ relevantes}{elementos\ extraidos}\ x\ 100$$

$$Precisión\ = \frac{24}{27}\ x\ 100$$

$$Precisión\ = \mathbf{88.89}\ \%$$

Como podemos apreciar se obtiene un error del 11.11%.

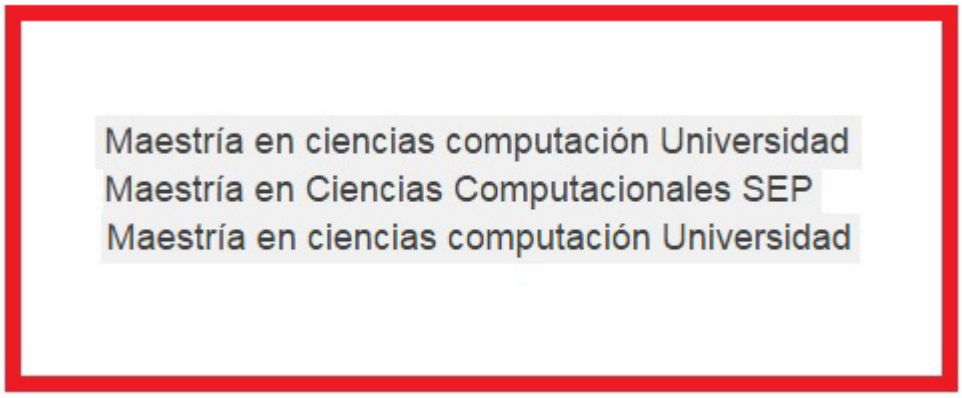


Figura 20. Tiempo estimado de procesamiento

7.2. Resultados de la extracción de experiencia profesional

Los resultados extraídos sobre la información de la experiencia profesional de los 12 expedientes curriculares, con un número total de 33 datos de experiencia profesional. El único inconveniente es que no todos los expedientes curriculares procesados tienen experiencia profesional, y un expediente profesional no tenía la estructura adecuada para aplicar las reglas semánticas y sintácticas empleadas.Con la ayuda de estos datos a continuación calcularemos la precisión del sistema web.

$$\boldsymbol{Elementos\ Erroneos\ Extraidos = 0}$$

$$\boldsymbol{Elementos\ Relevantes\ Extraidos = 33}$$

$$Precisión\ = \frac{elementos\ relevantes}{elementos\ extraidos}\ x\ 100$$

$$Precisión\ = \frac{33}{33}\ x\ 100$$

$$Precisión\ = \mathbf{100}\ \%$$

Como podemos apreciar se obtiene un error del 0%.

8. CONCLUSIONES

Como conclusión de este proyecto de integración podemos concluir que el objetivo primordial que era extraer información de aspectos académicos y experiencia profesional de expedientes curriculares descritos en idioma español aplicando reglas semánticas y sintácticas, se ha cumplido satisfactoriamente, sin embargo no toda la extracción es precisa porque en la técnica de extracción de aspectos académicos se encontraron deficiencias que tuvieron errores del 11.11% estos se pueden explicar debido a que el etiquetador utilizado presentaba algunas inconsistencias por ejemplo: algunos patrones empleados tomaban como nombre común o nombre propio palabras que no describían la información académica del expediente curricular, tal vez esto ocurra ya que dicho etiquetador esta optimizado para el idioma inglés. Sin embargo se logró una efectividad del 88.89% en la extracción de información de aspectos académicos.

En la extracción de la experiencia profesional no hubo índice de error en gran parte a los patrones empleados para su extracción, sin embargo no todos los expedientes curriculares en su estructura tenían un apartado de experiencia profesional y un expediente curricular en la experiencia profesional no tenía un patrón constante para poder aplicar las reglas semánticas y sintácticas.

Estos resultados son muy buenos ya que podemos optimizar el tiempo que emplea el ser humano en la búsqueda de información específica. En base a los resultados obtenidos podemos decir que es una herramienta confiable. Se considera que es notable este proyecto gracias a la importancia que tiene la extracción de información, porque es una herramienta de gran utilidad que sirve para la búsqueda de información en cantidades considerables de documentos.

Los resultados de este proyecto de integración pueden ser mejorados en trabajos posteriores, reestructurando algunas de las reglas semánticas y sintácticas implementadas, optimizando el código fuente de programación o utilizando otra herramienta como etiquetador.

9. ANEXOS

9.1. Código fuente de convertidor de PDF a TXT

```
/**
 *
 * @authorIvan Alejandro Rosas Torres
 */
public class PDFTxt {

PDFParserAnalizador;
   String AnalizarTexto;
PDFTextStripperPDFStripper;
PDDocumentPdDoc;
COSDocumentCosDoc;
PDDocumentInformationPddDocInf;

   //Extraer información del PDF
String PDFaText(String NombreArchivo){
       File inf = new File(NombreArchivo);
try {
Analizador = new PDFParser(new FileInputStream(inf));
Analizador.parse();
CosDoc = Analizador.getDocument();
PDFStripper = new PDFTextStripper();
PdDoc = new PDDocument(CosDoc);
AnalizarTexto = PDFStripper.getText(PdDoc);
}catch (IOException e){
}
returnAnalizarTexto;
   }
```

```
   //Escribir la información a un archivo
public void EscribirTxt(String PdfText, String NombreArchivo){
System.out.println("\nEscribiendo de PDF a TXT  " + NombreArchivo + "...");
try {
PrintWriter pw = new PrintWriter(NombreArchivo);
pw.println(PdfText);
pw.close();
}catch (IOException e){
       }
System.out.println("¡HECHO!");
}

//Extrae la información del PDF y se escribe en un TXT
public static String main(String PdfInf, String TxtInf) {
PDFTxtPdfTextAnaliz = new PDFTxt();
       String PdfaText = PdfTextAnaliz.PDFaText(PdfInf);

if (PdfaText == null) {

}else{
PdfTextAnaliz.EscribirTxt(PdfaText, TxtInf);
}
returnPdfaText;
   }
}
```

9.2. Código fuente de la limpieza de los expedientes curriculares

```
/**

 *

 * @authorIvan Alejandro Rosas Torres

 */

publicclassPalabrasVacias {


   Etiquetado etiq=new Etiquetado();


public String QuitarPalabras(String Ruta, String Nombre) throws IOException, TreeTaggerException, InstantiationException, IllegalAccessException{


BufferedReaderEntradaListaPalabras = null;

BufferedReaderEntradaPalabrasVacias = null;

StringLinea = "";

StringLineaAux= "";

StringTxtPalabrasVacias ="C:\\Users\\IART\\Desktop\\PT\\Documento\\PalabrasVacias.txt";

StringArchivoLimpio ="C:\\Users\\IART\\Desktop\\PT\\Documento\\limpieza\\";

ArrayList<String>PalabraVacia = new ArrayList<String>();

String[] temp=null;

   String delimiter = "_";


try{

int i = 0;

        File Archivo = new File(TxtPalabrasVacias);

FileReader r = new FileReader(Archivo);

EntradaListaPalabras = new BufferedReader(r);
```

```
        File Archivo2 = new File(Ruta);
FileReader r2 = new FileReader(Archivo2);
EntradaPalabrasVacias = new BufferedReader(r2);

        //Leer lista de palabras vacias
while((LineaAux = EntradaListaPalabras.readLine()) != null){
PalabraVacia.add(i, LineaAux);
i++;
        }
EntradaListaPalabras.close();
LineaAux="";

        //Leer palabras vacias
while((LineaAux = EntradaPalabrasVacias.readLine()) != null){
Linea = Linea + LineaAux + "\r\n" ;
        }
EntradaPalabrasVacias.close();
i = 0;

while (PalabraVacia.get(i) != null){
            //Linea = Linea.replaceAll(" ", "_");
            //Linea = Linea.replaceAll ("\\p{Punct}"," "); //Remuevesignos
            //Linea = Linea.replaceAll(" {2,}", " "); //Remueveespaciondobles
            //Linea = Linea.replaceAll ("\\d", " "); //Remuevenumeros
            Linea = Linea.replaceAll("\\s"+PalabraVacia.get(i)+"\\s"," "); //Remuevepalabravacia

i++;
```

```
            }
        } catch (Exception e) {}

    Linea=Linea.replace(".", " . ");
    Linea=Linea.replace(",", " , ");
    Linea=Linea.replace("-", " - ");
    Linea=Linea.replace(")", " ) ");
    Linea=Linea.replace(":", " : ");
    Linea= Linea.replaceAll("\\s","_");

     File documento = new File(ArchivoLimpio+Nombre);

try {
FileWriter w = new FileWriter(documento);
BufferedWriterbw = new BufferedWriter(w);
PrintWriterpr = new PrintWriter(bw);

temp = Linea.split(delimiter);

for(int i =0; i <temp.length ; i++){
pr.println(temp[i]);
            }
pr.close();
bw.close();
}catch(Exception e){}
String resultado = etiq.Etiquetar(temp);
```

```
return resultado;
  }
}
```

9.3. Código fuente de la segmentación y el etiquetado

```
publicclass Etiquetado {
   Doctorado doc = new Doctorado();
Ingenieriaing = new Ingenieria();
   Licenciatura lic =new Licenciatura();
Maestriamast = new Maestria();
   Experiencia exp1= new Experiencia();
Experiencia2 exp2 = new Experiencia2();

   private static ArrayList<String>Poss =new ArrayList<String>();
private static ArrayList<String>Tokenn =new ArrayList<String>();

public static ArrayList<String>getTokenn() {
returnTokenn;
   }

public static void setTokenn(ArrayList<String>Tokenn) {
Etiquetado.Tokenn = Tokenn;
   }

public static ArrayList<String>getPoss() {
returnPoss;
   }
```

```
public static void setPoss(ArrayList<String>Poss) {

Etiquetado.Poss = Poss;

   }


staticint x=1;

      String resultado= " ";

      String resultado2= " ";

      String resultado3= " ";

      String resultado4= " ";

      String resultado5= " ";

      String resultado6= " ";

      String resultado7= " ";

      String resultado8= " ";


public String Etiquetar(String [] Texto) throws IOException, TreeTaggerException,
InstantiationException, IllegalAccessException {


System.setProperty("treetagger.home", "C:/TreeTagger");

TreeTaggerWrapper<String>tt = new TreeTaggerWrapper<String>();


try {

tt.setModel("C:/TreeTagger/modelos/spanish.par:iso8859-1");

tt.setHandler(new TokenHandler<String>() {


@SuppressWarnings("empty-statement")

public void token(String token, String pos, String lemma) {

                        //System.out.println(pos + "\t" + token);
```

```
getPoss().add(pos);
getTokenn().add(token);
                    }
               }
                );
getPoss().clear();
getTokenn().clear();
tt.process(asList(Texto));

intidBD= consultas.AgregarDoc("Documento"+" "+ x);
resultado= "\n\nINFORMACION ACADEMICA: \n"+"<br />";
resultado2= "\n"+lic.Extractor(Poss, Tokenn, idBD)+"<br />";
          resultado3= "\n"+mast.Extractor(Poss, Tokenn, idBD)+"<br />";
          resultado4= "\n"+doc.Extractor(Poss, Tokenn, idBD)+"<br />";
resultado5= "\n"+ing.Extractor(Poss, Tokenn, idBD)+"<br /><br />";
          resultado6= "\n\nEXPERIENCIA PROFESIONAL:\n"+"<br />";
resultado7= "\n"+exp1.Extractor(Poss, Tokenn, idBD)+"<br />";
          resultado8= "\n"+exp2.Extractor(Poss, Tokenn, idBD)+"<br />";
}
finally {
tt.destroy();
      }
x++;
return resultado+resultado2+resultado3+resultado4+resultado5+resultado6+resultado7+resultado8;
   }
}
```

9.4. Código fuente de la extracción de información sobre aspectos académicos

```
**

 *

 * @authorIvan Alejandro Rosas Torres

 */

public class Licenciatura {


public String Extractor(ArrayList<String> Poss2,ArrayList<String> Tokenn2,int idBD) throws InstantiationException, IllegalAccessException{

      //ArrayList<String> T=Tokenn2;


int a = 0,f=0;

for (int i = 0; i <= Poss2.size()-1; i++) {


if((Tokenn2.get(i).equals("FORMACIÓN")&&Tokenn2.get(i+1).equals("ACADÉMICA"))

            ||(Tokenn2.get(i).equals("Grados")&&Tokenn2.get(i+1).equals("académicos"))


||Tokenn2.get(i).equals("Preparación Académica ")||Tokenn2.get(i).equals("Escolaridad")

            ||Tokenn2.get(i).equals("Grados académicos ")

            ){

          //System.out.println(T.get(i));

          a=i;

        }

if(Tokenn2.get(i).equals("RECONOCIMIENTOS")

            ||(Tokenn2.get(i).equals("Experiencia")&&Tokenn2.get(i+1).equals("profesional"))

            ||(Tokenn2.get(i).equals("Tesis")&&Tokenn2.get(i+1).equals("tesinas"))
```

```
||Tokenn2.get(i).equals("Ponencias y Artículos")||Tokenn2.get(i).equals("Tesis y tesinas realizadas "
)

                ){

                 // System.out.println(T.get(i));

                  f=i;

              }

       }


for (int i = a; i <=f; i++) {


if ((Tokenn2.get(i).equals("Licenciatura")||(Tokenn2.get(i).equals("Licenciado")))

&& (Tokenn2.get(i+1).equals("en"))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))

&&( Poss2.get(i+4).equals("NC")||Poss2.get(i+4).equals("NP"))

                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),
Tokenn2.get(i+3),Tokenn2.get(i+4),idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3) + " "
+ Tokenn2.get(i+4);


        } else if ((Tokenn2.get(i).equals("Licenciatura")||(Tokenn2.get(i).equals("Licenciado")))

&& (Tokenn2.get(i+1).equals("en"))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))

                ){
```

```
consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2), Tokenn2.get(i+3),"",idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3);


        } else if ((Tokenn2.get(i).equals("Licenciatura")||(Tokenn2.get(i).equals("Licenciado")))
&& (Tokenn2.get(i+1).equals("en"))
&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP")||Poss2.get(i+2).equals("ADJ"))
){
consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2), "","",idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2);
}else{
          }
      }
return "";
   }
}


/**
 *
 * @authorIvan Alejandro Rosas Torres
*/
public class Maestria {


public String Extractor(ArrayList<String> Poss2,ArrayList<String> Tokenn2,int idBD) throws InstantiationException, IllegalAccessException{
      //ArrayList<String> T=Tokenn2;
```

```
int a = 0,f=0;

for (int i = 0; i <= Poss2.size()-1; i++) {


if((Tokenn2.get(i).equals("FORMACIÓN")&&Tokenn2.get(i+1).equals("ACADÉMICA"))


||Tokenn2.get(i).equals("Grados Académicos ")||Tokenn2.get(i).equals("ESCOLARIDAD")


||Tokenn2.get(i).equals("Preparación Académica ")||Tokenn2.get(i).equals("Escolaridad")

                ){

            //System.out.println(T.get(i));

            a=i;

        }

if(Tokenn2.get(i).equals("RECONOCIMIENTOS")||Tokenn2.get(i).equals("Reconocimientos y Mem
bresías  ")


||(Tokenn2.get(i).equals("PRODUCTOS")&&Tokenn2.get(i+1).equals("INVESTIGACIÓN"))

              ||Tokenn2.get(i).equals("Ponencias y Artículos")

              ||(Tokenn2.get(i).equals("Experiencia")&&Tokenn2.get(i+1).equals("profesional"))

              ){

                //System.out.println(T.get(i));

                f=i;

            }




    }

for (int i = a; i <=f; i++) {
```

```
if ((Tokenn2.get(i).equals("Maestría")||(Tokenn2.get(i).equals("Maestro")))

&& (Tokenn2.get(i+1).equals("en"))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))

&&( Poss2.get(i+4).equals("NC")||Poss2.get(i+4).equals("NP"))

                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),
Tokenn2.get(i+3),Tokenn2.get(i+4),idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3) + " "
+ Tokenn2.get(i+4);


}else  if ((Tokenn2.get(i).equals("Maestría")||(Tokenn2.get(i).equals("Maestro")))

&& (Tokenn2.get(i+1).equals("en"))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))

                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),
Tokenn2.get(i+3),"",idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3);


        } else if ((Tokenn2.get(i).equals("Maestría")||(Tokenn2.get(i).equals("Maestro")))

&& (Tokenn2.get(i+1).equals("en"))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),"","",idBD);
```

```
return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2);

}else{

          }

       }

return "";

   }

}


/**

 *

 * @authorIvan Alejandro Rosas Torres

*/

public class Doctorado {

public String Extractor(ArrayList<String> Poss2,ArrayList<String> Tokenn2, intidBD) throws InstantiationException, IllegalAccessException{

       //ArrayList<String> T=Tokenn2;


int a = 0,f=0;

for (int i = 0; i <= Poss2.size()-1; i++) {


if((Tokenn2.get(i).equals("FORMACIÓN")&&Tokenn2.get(i+1).equals("ACADÉMICA"))


||Tokenn2.get(i).equals("Grados Académicos ")||Tokenn2.get(i).equals("ESCOLARIDAD")


||Tokenn2.get(i).equals("Preparación Académica ")||Tokenn2.get(i).equals("Escolaridad")

                ||(Tokenn2.get(i).equals("Estudios")&&Tokenn2.get(i+1).equals(":"))

                ){

            //System.out.println(T.get(i));
```

```
            a=i;
          }
if(Tokenn2.get(i).equals("RECONOCIMIENTOS")||Tokenn2.get(i).equals("Reconocimientos y Mem
bresías  ")

||(Tokenn2.get(i).equals("PRODUCTOS")&&Tokenn2.get(i+1).equals("INVESTIGACIÓN"))
                ||Tokenn2.get(i).equals("Ponencias y Artículos")
                ||(Tokenn2.get(i).equals("Experiencia")&&Tokenn2.get(i+1).equals("profesional"))
                ){
            // System.out.println(T.get(i));
            f=i;
            }
      }

for (int i = a; i <=f; i++) {

if ((Tokenn2.get(i).equals("Doctorado")||(Tokenn2.get(i).equals("Doctor")))
&& (Tokenn2.get(i+1).equals("en"))
&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))
&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))
&&( Poss2.get(i+4).equals("NC")||Poss2.get(i+4).equals("NP"))
                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),
Tokenn2.get(i+3),Tokenn2.get(i+4),idBD);

return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3) + " "
+ Tokenn2.get(i+4);
```

```
}else if ((Tokenn2.get(i).equals("Doctorado")||(Tokenn2.get(i).equals("Doctor")))
&& (Tokenn2.get(i+1).equals("en"))
&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))
&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))
                //||Poss2.get(i+3).equals("ADJ"))
                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2), Tokenn2.get(i+3),"",idBD);

return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3);

}else if ((Tokenn2.get(i).equals("Doctorado")||(Tokenn2.get(i).equals("Doctor")))
&& (Tokenn2.get(i+1).equals("en"))
&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))
){
consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2), "","",idBD);

return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2);
}else{
          }
       }
return "";
   }
}

/**
 *
```

```
 * @authorIvan Alejandro Rosas Torres

*/

public class Ingenieria {

public String Extractor(ArrayList<String> Poss2,ArrayList<String> Tokenn2,int idBD) throws InstantiationException, IllegalAccessException{

      //ArrayList<String> T=Tokenn2;


int a = 0,f=0;

for (int i = 0; i <= Poss2.size()-1; i++) {


if(Tokenn2.get(i).equals("Grados Académicos ")||Tokenn2.get(i).equals("ESCOLARIDAD")){

            //System.out.println(T.get(i));

            a=i;

         }

if(Tokenn2.get(i).equals("Reconocimientos y Membresías  ")


||(Tokenn2.get(i).equals("PRODUCTOS")&&Tokenn2.get(i+1).equals("INVESTIGACIÓN"))

               ){

                 //System.out.println(T.get(i));

                 f=i;

            }

      }

for (int i = a; i <=f; i++) {


if ((Tokenn2.get(i).equals("Ingeniería")||(Tokenn2.get(i).equals("Ingeniero")))

&& (Tokenn2.get(i+1).equals("en"))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))
```

```
&&( Poss2.get(i+4).equals("NC")||Poss2.get(i+4).equals("NP"))

                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),
Tokenn2.get(i+3),Tokenn2.get(i+4),idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3) + " "
+ Tokenn2.get(i+4);


        }else                                                                   if
((Tokenn2.get(i).equals("Ingeniería")||(Tokenn2.get(i).equals("Ingeniero"))||(Tokenn2.get(i).equals("I
ng")))

&& ((Tokenn2.get(i+1).equals("en"))||(Tokenn2.get(i+1).equals(".")))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

&&( Poss2.get(i+3).equals("NC")||Poss2.get(i+3).equals("NP"))

                ){


consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),
Tokenn2.get(i+3),"",idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2) + " " + Tokenn2.get(i+3);


}else if ((Tokenn2.get(i).equals("Ingeniería")||(Tokenn2.get(i).equals("Ingeniero")))

&& (Tokenn2.get(i+1).equals("en"))

&&( Poss2.get(i+2).equals("NC")||Poss2.get(i+2).equals("NP"))

                ){

consultas.AgregarEscolaridad(Tokenn2.get(i), Tokenn2.get(i+1), Tokenn2.get(i+2),"","",idBD);


return Tokenn2.get(i) + " " + Tokenn2.get(i+1)+ " "+ Tokenn2.get(i+2);

}else{
```

```
        }
      }
return "";
   }
}
```

9.5. Código Fuente de la extracción de experiencia profesional

```
/**
 *
 * @authorIvan Alejandro Rosas Torres
 */
public class Experiencia {


public String Extractor(ArrayList<String> Poss2,ArrayList<String> Tokenn2, intidBD) throws InstantiationException, IllegalAccessException{
      //ArrayList<String> T=Tokenn2;


int a = 0,f=0;
for (int i = 0; i <= Poss2.size()-1; i++) {


if((Tokenn2.get(i).equals("INVESTIGACIÓN")&&Tokenn2.get(i+1).equals("DOCENCIA"))
                ||(Tokenn2.get(i).equals("Experiencia")&&Tokenn2.get(i+1).equals("profesional"))
                ){
            //System.out.println(T.get(i));
            a=i;
        }
if((Tokenn2.get(i).equals("TESIS")&&Tokenn2.get(i+1).equals("MAESTRÍA"))
```

```
                ||(Tokenn2.get(i).equals("Artículos")&&Tokenn2.get(i+1).equals("investigación"))
                ){
                  //System.out.println(T.get(i));
                  f=i;
              }
        }

        String s=" ";
        String s2="";

for (int i = a+2; i <=f; i++) {

        s=s+Tokenn2.get(i)+ " ";

if((Tokenn2.get(i).equals(".")&&((Poss2.get(i-1).equals("CODE"))||(Poss2.get(i-
1).equals("CARD"))))||
              (Tokenn2.get(i).equals(".")&&((Poss2.get(i-1).equals("ADJ"))&&(Tokenn2.get(i-
1).equals("indefinido"))))
){

consultas.AgregarExperiencia(s,idBD);

                  s2+=s+"\n"+"<br />";
s=" ";
               }
           }
return s2;
    }
```

```
}


/**
 *
 * @author Ivan Alejandro Rosas Torres
 */
public class Experiencia2 {


public String Extractor(ArrayList<String> Poss2,ArrayList<String> Tokenn2, intidBD) throws InstantiationException, IllegalAccessException{
      //ArrayList<String> T=Tokenn2;


int a = 0,f=0;
for (int i = 0; i <= Poss2.size()-1; i++) {


if((Tokenn2.get(i).equals("Experiencia")&&Tokenn2.get(i+1).equals("profesional"))

||(Tokenn2.get(i).equals("INVESTIGACIÓN")&&Tokenn2.get(i+1).equals("DOCENCIA"))
                 ){
            //System.out.println(T.get(i));
            a=i;
         }
if((Tokenn2.get(i).equals("Posición")&&Tokenn2.get(i+1).equals("categoría"))
               ||(Tokenn2.get(i).equals("TESIS")&&Tokenn2.get(i+1).equals("MAESTRÍA"))
               ||(Tokenn2.get(i).equals("Artículos")&&Tokenn2.get(i+1).equals("investigación"))
               ){
                 //System.out.println(T.get(i));
                 f=i;
```

```
            }
        }

        String s=" ";
        String s2="";

for (int i = a+2; i <=f; i++) {

        s=s+Tokenn2.get(i)+ " ";

if                          (Tokenn2.get(i).equals(".")&&(Poss2.get(i-1).equals("NP"))&&(Tokenn2.get(i-
2).equals(",")||(Tokenn2.get(i-2).equals(".")))
                    ||(Poss2.get(i).equals("RP")&&(Poss2.get(i-1).equals("VLadj")))
){

consultas.AgregarExperiencia(s,idBD);

                s2+=s+"\n"+"<br />";
                s=" ";
              }
          }
return s2;
}
}
```

9.6. Código fuente del almacenamiento de la información extraída

```
/**
```

```
 *
 * @authorIvan Alejandro Rosas Torres
 */

importjava.sql.Connection;
importjava.sql.ResultSet;
importjava.sql.SQLException;
importjava.sql.Statement;

public class consultas  {

public  staticintAgregarDoc(String Doc) throws InstantiationException, IllegalAccessException{

int m = 0;

        try {
                Conexion c=new Conexion();
                Connection con=c.getConexion();
                if(con!=null){
                        Statement st;
                        st = con.createStatement();
ResultSet r;
                    String r1 = null;

st.executeUpdate("insert into pdf (Documento) values (\""+Doc+"\");");
                     r = st.executeQuery("SELECT MAX(IDdoc) AS id FROM pdf;");
while(r.next())
```

```
                    {
                        r1=r.getString("id").toString();
                    }
                m=Integer.parseInt(r1);

        st.close();

}
                c.cerrarConexion();
        } catch (SQLException e) {
System.out.println("error"+e);
        }
return m;
   }

    public  static void  AgregarEscolaridad(String Tokenn1,String Tokenn2,String Tokenn3,String Tokenn4,String Tokenn5, intidBD) throws InstantiationException, IllegalAccessException{
        String esp=" ";
        try {
                Conexion c=new Conexion();
                Connection con=c.getConexion();
                if(con!=null){
                    Statement st;
                    st = con.createStatement();

st.executeUpdate("insert into escolaridad (Informacion, IDdoc) values (\""+Tokenn1+esp+Tokenn2+esp+Tokenn3+esp+Tokenn4+esp+Tokenn5+"\",\""+idBD+"\");");
```

```
            st.close();


                    }

                     c.cerrarConexion();

            } catch (SQLException e) {

System.out.println("error"+e);

            }

      }


public static void AgregarExperiencia(String s, intidBD) throws InstantiationException, IllegalAccessException{


            try {

                        Conexion c=new Conexion();

                        Connection con=c.getConexion();

                        if(con!=null){

                                    Statement st;

                                    st = con.createStatement();


st.executeUpdate("insert into experiencia (Descripcion, IDdoc) values (\""+s+"\",\""+idBD+"\");");

            st.close();

                    }

                     c.cerrarConexion();

            } catch (SQLException e) {

System.out.println("error"+e);

            }

      }

}
```

9.7. Código fuente del módulo de validación

```
public class indexControler implements Serializable{

publicindexControler(){}

private final Set<File>archivos = new HashSet<File>(0);

private List<String>nomArchivos = new ArrayList<String>(0);

/*Habilitar y deshabilitar botones de la vista*/

privatebooleanbtnIniciar = false, btnPoblar = true, btnCancelar = true, btnFinalizar = true;

   /*Directorio temporal para guardar los archivos subidos al servidor*/

private File tempDirColeccion;

   /*resultados a mostrar en la vista*/

private String resultados;


private Integer progress = 0, tmpProgress = 0;

privateintopcionSel, sizeNum = 0, iTam = 0;


publicvoidiniciarProceso(){

opcionSel = 1;

btnPoblar=btnIniciar = true;

btnCancelar = false;

progress = 0;

FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "Cargarexpedientescurriculares.", null));

   }


public void cancelarProceso(){

terminarProceso();
```

```
FacesContext.getCurrentInstance().addMessage(null, new
FacesMessage(FacesMessage.SEVERITY_INFO, "Procesocancelado.", null));

}


publicvoidterminarProceso(){

btnIniciar = false;

btnPoblar = btnCancelar = btnFinalizar = true;

opcionSel = progress = iTam = 0;

resultados = "";

borrarArchivosTemporales();

}


public void subirArchivos(FileUploadEvent event) throws IOException{

procesaArchivos(1,event);

   }


public void subirArchivosB(FileUploadEvent event) throws IOException{

procesaArchivos(2,event);

   }


public void subirArchivosC(FileUploadEvent event) throws IOException{

procesaArchivos(3,event);

   }


public void procesaArchivos(int x, FileUploadEvent event) throws IOException{

//Si no hay un directorio temporal, se crea un directorio temporal en la carperaTemp

if(tempDirColeccion == null){

        File tempDirUser = new File(System.getProperty("java.io.tmpdir"));
```

```
FacesContextctx = FacesContext.getCurrentInstance();
HttpSession Session = (HttpSession)(ctx.getExternalContext().getSession(false));
        String id = Session.getId();
tempDirColeccion = new File(tempDirUser, id);
if (!tempDirColeccion.exists()) tempDirColeccion.mkdir();
tempDirColeccion.deleteOnExit();
      }
      String filename = FilenameUtils.getBaseName(event.getFile().getFileName());
      String extension = FilenameUtils.getExtension(event.getFile().getFileName());
      File archivoTmp = new File(tempDirColeccion,filename+"."+extension);
InputStream stream = event.getFile().getInputstream();
IOUtils.copy(stream, new FileOutputStream(archivoTmp));
if(x==1){
if(archivos.add(archivoTmp)) nomArchivos.add(archivoTmp.getName());
}
      //Mientras no se agreguen archivos fuente y destino, el boton comparar permanece
deshabilitado//
btnPoblar=false;


   }


privatevoidborrarArchivosTemporales(){
int i;
if(!(archivos.isEmpty()) ){
archivos.clear();   nomArchivos.clear();
File[] a = tempDirColeccion.listFiles();
for (i=0; i<a.length; i++)
a[i].delete();
```

```
tempDirColeccion.delete();
tempDirColeccion = null;
        }
    }

publicbooleanisBtnCancelar(){
returnbtnCancelar;
    }

public void setBtnCancelar(booleanbtnCancelar) {
this.btnCancelar = btnCancelar;
    }

publicbooleanisBtnFinalizar() {
returnbtnFinalizar;
    }

public void setBtnFinalizar(booleanbtnFinalizar) {
this.btnFinalizar = btnFinalizar;
    }

publicbooleanisBtnIniciar() {
returnbtnIniciar;
    }

public void setBtnIniciar(booleanbtnIniciar) {
this.btnIniciar = btnIniciar;
```

```
    }

publicbooleanisBtnPoblar() {
returnbtnPoblar;
    }

public void setBtnPoblar(booleanbtnPoblar) {
this.btnPoblar = btnPoblar;
    }

publicintgetOpcionSel() {
returnopcionSel;
    }

public void setOpcionSel(intopcionSel) {
this.opcionSel = opcionSel;
    }

public List<String>getNomArchivos() {
returnnomArchivos;
    }

public void setNomArchivos(List<String>nomArchivos) {
this.nomArchivos = nomArchivos;
    }

publicStringgetResultados() {
```

```
return resultados;
   }

publicvoidsetResultados(String resultados) {
this.resultados = resultados;
   }

   /*Metodo para controlar la barra de progreso*/
public Integer getProgress() {
if(sizeNum<1) sizeNum=1;
if(progress == null){
progress = 0;
      }
else{
tmpProgress=(int)((100*iTam)/(sizeNum));
if(tmpProgress>=99 &&iTam<sizeNum) progress=99;
else progress=tmpProgress;
      }
return progress;
   }

public void setProgress(Integer progress) {
this.progress = progress;
   }

public void muestraPdf() throws InstantiationException, IllegalAccessException, IOException,
TreeTaggerException{
```

```
resultados="<big><b>Resultados</big></b>";


String CarpetaDoc = tempDirColeccion.toString()+"\\";
   String CarpetaTxt = "C:\\Users\\IART\\Desktop\\PT\\Documento\\txt\\";
StringCarpetaLimpia = "C:\\Users\\IART\\Desktop\\PT\\Documento\\limpieza\\";


PDFTxt p = new PDFTxt();
PalabrasVaciaspv = new PalabrasVacias();
EliminarArchivoelim = new EliminarArchivo();
EliminarTxtPdf elim2 = new EliminarTxtPdf();


        File pdf = new File(CarpetaDoc);
        File arch = new File(CarpetaLimpia);
        File doc = new File (CarpetaTxt);


if (pdf.exists())
      {
File[] ficheros = pdf.listFiles();
for (int x=0;x<ficheros.length;x++)
          {
p.main(CarpetaDoc + ficheros[x].getName(), CarpetaTxt +
ficheros[x].getName().replace(".pdf",".txt"));


          }
      }
        String resultadofinal;
        File f2 = new File(CarpetaTxt);
```

```
File[] ficheros2 = f2.listFiles();
int y=1;
for (int x=0;x<ficheros2.length;x++){
resultados+="<br/></br>"+"\nDocumento "+y+"<br />";
resultadofinal = pv.QuitarPalabras(CarpetaTxt+ficheros2[x].getName(), ficheros2[x].getName());
System.out.println(resultadofinal);
resultados+="<br/></br>"+resultadofinal;
y++;
        }

if (arch.exists())
      {
File[] ficheros4 = arch.listFiles();
for (int x=0;x<ficheros4.length;x++)
{
elim.EliminarTxt(CarpetaLimpia + ficheros4[x].getName());
}
      }

if (doc.exists())
      {
File[] ficheros5 = doc.listFiles();
for (int x=0;x<ficheros5.length;x++)
{
elim2.EliminarDocumento(CarpetaTxt + ficheros5[x].getName());
           }
      }
```

```
btnCancelar = btnPoblar = true;


btnFinalizar = false;

opcionSel = 0;

borrarArchivosTemporales();

 }


}
```

10. BIBLIOGRAFIA

[1] J. Pascual Martínez, "*Extracción automatizada y representación de servicios Web mediante ontologías",* proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[2] S.M. Ugalde Chávez, "*Sistema de recuperación de información semántico",* proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2012.

[3] F. Tébar Martínez, *"Sistema de almacenamiento semántico y recuperación de textos de investigación mediante ontologías"*, proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2013.

[4] H. Aguilar Galicia, "*Extracción automática de información semántica basada en estructuras sintácticas"*, tesis de maestría, Centro de Investigación en Computación, Instituto Politécnico Nacional, México, 2010.

[5] N.Kushman, Y. Artzi, L. Zettlemoyer, y R.Barzilay *,"Learning to Automatically Solve Algebra Word Problems"*, 2014.

[6] Anthony, L**.** "*AntConc in Action: Using Corpus Linguistics Tools and Techniques to Investigate Morphology, Syntax, Semantics, Pragmatics, and Language Variation*". 2nd Korea Association of Corpus Linguistics Conference, Korea University, Seoul, Korea, 2013.